

Servlet Container Jetty

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de servlet. Y en este caso vamos a hacer un deploy nuestra aplicación. ¿Pero como así Bruno? Si nosotros ya hicimos un deploy de nuestra aplicación en Tomcat. Bueno, lo que vamos a hacer es un deploy pero en otro Servlet container.

[00:21] Entonces, antes de continuar con ese deploy vamos a ver qué es un servlet container. Tomcat es un Servlet container. ¿Qué es lo que significa? Lo que hace es hacer un mapeo entre una URL y un objeto Java. Entonces básicamente hace eso, es bastante simple y pero efectivo.

[00:43] Jetty por ejemplo es otro servel container, es otro tipo, es otro servidor del mismo tipo. También hace ese mapeamiento de URL a objetos. Tal vez Tomcat es el más conocido entre ellos dos, pero Jetty también es bastante conocido y utilizado. Existe otro tipo de servidor. En este caso serían los application Server, y de esos application servers yo traje una lista de tres.

[01:18] De entre todos los que existen, creo que son los más conocidos, tenemos este Wildfly por ejemplo, que es el más conocido de todos, es un servidor gratuito. Tenemos weblogic que es un servidor pago, es de Oracle Wildfly es de red hat. Y websphere es de IBM también es pago. Normalmente estos tienen un costo alto, y Wildfly lo que se hace es pagar por el soporte normalmente.

[01:46] Pero ahora, ¿cuál es la diferencia entre un servlet container o un Servlet engine y un application server? Bien, la diferencia es que una application server más tiene muchas más funcionalidades. Por ejemplo ya trae varias bibliotecas que nosotros podemos utilizar directamente sin tener que instalar nada extra, entonces trae, por ejemplo, bibliotecas para convertir a JSON XML, algunos generadores de base de datos digamos.

[02:15] Trae también herramientas para poder hacer un buen control de las transacciones de los datos, entonces es bastante más completo, perdón, que un servlet container. Pero la contra que tiene es que también es más pesado, digamos, un Servlet container puede pesar Tomcat, creo que estaba en unos 15 20 megas, si no me equivoco, y application server puede pesar, no sé, 80 megas tal vez más.

[02:45] Entonces son más pesados. Como les dije, estos son los más conocidos de los application server, Wildfly es el más conocido. De server containers, Tomcat es el más conocido. Pero bueno, basta de aplicaciones, vamos a ir al código, vamos a implementar un servidor Jetty y vamos a transformar. A transformar no, a llevar nuestra aplicación que hemos creado a un servidor diferente, que en este caso es Jetty.

[03:17] Entonces para eso vamos a ir a nuestro código y vamos a exportar nuestro gerenciador. Nosotros ya lo habíamos hecho anteriormente, vamos a hacer un clic derecho en nuestro gerenciador export y acá si ustedes no les aparece el WAR file, vayan a export y acá colocan War, ahí va a aparecer WAR file. Siguiente.

[03:44] Y acá vamos a colocar, vamos a decir dónde lo vamos a colocar. En mi caso, yo lo voy a colocar cerca de donde tengo todas las otras cosas instaladas, Apache, Eclipse y todo eso. Voy a guardar esa ubicación. Destilden esa opción de optimizar para el Servlet Apache, porque nosotros lo vamos a utilizar en otro servidor, entonces no coloquen esa optimización.

[04:10] Y si no tienen puesto coloquen ese `overwrite existing file`, solo por las dudas. Vamos a dar un finish. En teoría, ya deberíamos tener nuestro gerenciador WAR y ahora lo que falta es bajar nuestro Jetty. Vamos a ir al Google Chrome o a cualquier navegador. Busquen Jetty. El primer resultado que le aparece va a ser la página oficial, seguramente, es de la Eclipse Foundation, vamos a hacer clic acá.

[04:45] Y acá, por ejemplo, tiene la opción de descargarlo. Yo en este caso yo ya descargué la opción 11.09. Les recomiendo mucho que utilicen la misma versión que yo estoy utilizando porque entre versión y versión cambian las formas de poder inicializar Jetty.

[05:03] Entonces yo ya lo tengo bajado y bueno, y se los he compartido también dentro de acá de Alura. Y si queremos saber cómo iniciar ese Jetty, acá tenemos la documentación, si nos vamos a documentación, Operations Guide, y acá en getting started tenemos ese starting Jetty. Acá nos está diciendo que básicamente tenemos que extraer ese archivo en algún lugar que no tenga espacios, perdón en un path, en un camino que no tenga espacio, en carpetas que no tengan espacios.

[05:42] Y vamos a hacer, vamos a seguir esta guía. Vamos a ir, yo lo tengo acá en descargas, Jetty-home, yo voy a extraerlo acá. Voy a extraer lo clic derecho, extraer en jetty-home. Acá tengo la carpeta, tenemos este, no sé por qué apareció dos veces. Voy a sacar esto. Bien, ahora sí, ahora sí apareció una sola vez, ese jetty-home.

[06:08] Voy a cortarlo, clic derecho, cortar o "Ctrl + X", voy a llevarlo también a mi carpeta acá, que yo tengo de Servlet, donde tengo mi gerenciador WAR y todo eso, voy a hacer un "Ctrl + V". Esperamos que se termine de copiar. Perfecto. Entonces tenemos ese jetty-home, ¿qué es lo que ocurre? Acá nos está diciendo en esta guía que ese jetty-home nosotros no tenemos que tocarlo.

[06:35] Entonces lo que sea que tenga adentro todos estos archivos, nosotros no tenemos que crear nada, nosotros al final vamos a terminar ejecutando ese start.jar, pero por ahora nuestro generador no va a ir a ese jetty-home, sino que vamos a tener que crear una nueva carpeta que acá en la documentación les llamaron de jetty.base.

[06:56] A partir de jettyBase ahí nosotros vamos a colocar todo nuestro proyecto en ese jettyBase y vamos a ejecutar ese start de jetty-home desde dentro de nuestra carpeta jettyBase. ¿Cómo vamos a hacer eso? Vamos a ver. En esta parte de acá, en la misma altura, digamos que un jetty-home, vamos a crear una nueva carpeta y yo le voy a llamar de jettyBase.

[07:22] Voy a colocarlo todo junto porque nos habían dicho ahí que no coloquemos espacios, entonces vamos a hacer eso, y acá tenemos nuestro jettyBase sin nada dentro. ¿Qué es lo que vamos a hacer? Nosotros vamos a hacer un ciudad, vamos a primero abrir nuestro command símbolo del sistema.

[07:40] Este caso yo estoy ubicado en C:\Users\Alura, pero tengo que ir a mi disco D, que es donde yo tengo mi jettyBase. Este es mi camino, entonces voy a copiar desde esta parte después de mi disco, que es el camino bien y dentro de mi símbolo del sistema yo voy a hacer un D: porque voy a ir a mi disco D. Enter ahora estoy parado en mi disco D, voy a hacer un cd, espacio y voy a hacer un clic derecho para pegar lo que yo había copiado.

[08:12] Esto que yo copie acá. En este caso es trabajo\Alura\Servlet\jettyBase, etcétera, copiar, y voy a hacer un pegar. "Ctrl + V", no lo copié bien. A ver un segundo. "Ctrl + C", "Ctrl + V". Se pegó dos veces, problemas técnicos. JettyBase, hasta acá. Perfecto, si ven, estoy parado ahora en jettyBase. Estoy parado acá, ¿ahora qué es lo que nosotros tenemos que hacer?

[08:48] En ese jettyBase, en esa carpeta, nosotros vamos a comenzar a importar módulos. Jetty viene por default, así de forma inicial, con pocos módulos, en

realidad, con pocos módulos. Nosotros vamos a tener que instalar un módulo de deploy y uno de JSP. El de deploy va a ser para poder colocar cualquier aplicación web, colocarla ahí, cualquier .WAR, colocarlo dentro de la carpeta Webapp.

[09:17] Y el JSP es porque específicamente nuestro proyecto utiliza JSP y no viene dentro del módulo de deploy, entonces vamos a instalar esos módulos. ¿Cómo sé cuáles son los módulos y cómo instalarlos? Acá en esta parte de la documentación nos está diciendo que tenemos que utilizar esos comandos con esos parámetros para poder ver todos los módulos.

[09:46] Si jetty-home es el camino hacia nuestro jetty-home que nosotros colocamos. En este caso mi jetty-home está acá. Entonces voy a abrir ese jetty-home solo para copiar su camino. "Ctrl + C", hasta acá, bien. En realidad creo que no lo necesito, vamos a ir a así, vamos a ir a nuestro jettyBase, yo tengo acá, estoy en mi símbolo del sistema y vamos a hacer lo siguiente, voy a copiar esto. Copiar.

[10:20] Perdón que tenía "Ctrl + C", "Ctrl + V". Y acá en vez de jetty-home, voy a hacer lo siguiente. Si ustedes hicieron lo que yo les había dicho de colocar la carpeta jettyBase en la misma altura que jetty-home, ustedes tienen que usar el siguiente comando: ./, tengo el en español acá en inglés: ., esta es también la barra invertida.

[10:50] \ que significa que voy a ejecutar una cosa .\ significa que voy a volver para atrás. ¿Qué es lo que ocurre? Yo estoy parado, no es el foco de este curso, pero vamos a explicar. Yo estoy parado en jettyBase. Con el \ dije "voy a ejecutar una cosa". ¿Qué cosa? Tengo que localizar esa cosa que voy a ejecutar, ese start.jar. Entonces lo que hago es usar .\ para volver para atrás, ahora estoy posicionado en Servlet.

[11:20] Y ahora tengo que ir a mi jetty-home, entonces después de esa barra vamos a colocar jetty-home, perfecto, barra y acá vamos a usar ese start.jar, ya

que ahora estamos parados dentro de esta carpeta jetty-home. Entonces podemos localizar ese start.jar. Entonces colocamos start, si aprietan la tecla tab les va a autocompletar, lo mismo con el nombre de jetty-home, espacio, y acá nos habían dicho que teníamos que colocar este parámetro -list-modules. Copiamos, pegamos y con esto vamos a ver todos los módulos que podemos instalar.

[12:06] Son muchos. Acá por ejemplo tenemos el JSP que es para habilitar los JSP y en nuestra web application y nosotros vamos a usar también el módulo deploy que es para deployar una web, un web application. Entonces una vez que vimos esas posibilidades, vamos estando acá parados en el prompt, vamos a hacer un clic con la flecha para arriba, para repetir esto, pero no vamos a repetir ese mismo comando.

[12:41] Vamos a usar otro comando que aparece acá en la documentación, llamado add-module para agregar un módulo. Cada una de todas esas listas tenía módulos. Entonces vamos a hacer un add-module, no me acuerdo si es en singular, add-module = y vimos que se llamaba deploy uno de los que necesitamos, enter.

[13:06] Perfecto, nos está diciendo acá que consiguió agregarlo y vamos a hacer lo mismo, pero para el módulo JSP. Enter. Con eso si no me engaño, ya tendríamos todo lo necesario para poder colocar nuestro archivo WAR dentro de ese base, vamos a ver cómo es que quedó ese jetty-base.

[13:26] Volvemos a nuestro Servlet, que es donde yo tenía mi jetty-base, doble clic y vean que tenemos estas carpetas. Veán que tenemos acá una carpeta webapps, acá es donde vamos a colocar nuestro .war. Entonces voy a mover mi gerenciaador.war, lo mando a jetty-base y de gerenciador.war, lo mando a webapps, tenemos acá dentro de nuestro .war.

[13:50] Con esto, ya finalizada esa configuración, nosotros ya podríamos ejecutar Jetty, y vamos a ver si funciona. Voy a hacer, hice clic en la flecha para

arriba para poder ejecutar el último comando, pero voy a eliminar el parámetro, entonces nos va a quedar ese con ese start.jar. Enter y vamos a ver si funciona.

[14:15] No parece que haya ningún error, vamos a ver si está funcionando. Yo mi servlet acá está parado, está stopped. Entonces vamos a ver qué es lo que pasa, localhost:8080, quiero ver primero, qué es lo que tenemos acá. Localhost:8080. Se está ejecutando. Creo que ha faltado que importe un módulo, entonces voy a hacer un "Ctrl + C" acá, para finalizar nuestra ejecución voy a subir dos veces con la flechita para arriba y voy a agregar el módulo HTTP.

[14:48] Creo que ese es el que me estaba faltando. Vamos a hacer de nuevo Jetty start, enter. Vamos a ver ahora si funciona. Ahora funcionó. Tenemos nuestro Jetty, que nos está diciendo 404 not found, en un localhost:8080. Eso está todo bien, pero sabemos que fue Jetty quien nos respondió, entonces lo que vamos a hacer ahora sí es ir a nuestro gerenciador/ListaEmpresas. Hice una búsqueda en Google.

[15:23] Enter, vamos a ver qué pasa. Tenemos nuestro gerenciador, vamos a ver si funciona. Bruno 1345, enviar, bien. Pudimos logarnos, vamos a ver si podemos modificar, 123 enviar podemos modificar, eliminar, perfecto. Entonces nuestro servlet está funcionando. ¿Ahora qué es lo que deberíamos probar? Si nosotros tenemos fíjense que acá en la consola de jetty apareció todo los comandos que hemos ejecutado.

[15:55] Ahora en teoría nuestro cliente webservice debería funcionar también, porque simplemente hemos levantado un servidor. No importa si es Tomcat o si es Jetty, nuestro cliente debería poder acceder a esa URL, entonces vamos a hacer clic en clienteWebService y vamos a hacer clic derecho run as Java application. A ver.

[16:16] Bien, nos devolvió un xml, yo tenía acá xml, vamos a intentar con JSON "Ctrl + S" y vamos a ejecutarlo de nuevo. Perfecto, nos devolvió un solo item. ¿Por qué? Porque nosotros habíamos eliminado un objeto, una empresa entonces, por eso nos devolvió solamente un item. Entonces esto está funcionando.

[16:38] Hemos conseguido hacer un deploy de Jetty. ¿Qué es lo que esto significa? Significa que ahora nuestro servidor, nuestro servidor no, nuestra aplicación web podemos nosotros llevarla, por ejemplo, si quisiéramos a Wildfly. por ejemplo o Weblogic o Websphere. cualquiera de ellos bien. va que